

Gerador de Instancias - TP GCC218

October 28, 2018

1 Gerador de Instâncias

- Conjuntos elementares

- N : número de clientes a serem atendidos no dia;
- R : número de sub-regiões;
- K : quantidade de tipos diferentes de veículos disponíveis;
- H : número de horas da jornada diária.

```
In [16]: N = 100
         R = 5
         K = 5
         H = 7
```

- Dados a respeito de um cliente $i \in N$

- v_i : volume (em m^3) do pedido a ser entregue;
- p_i : valor (em reais) do pedido a ser entregue;
- n_i : quantidade de pacotes que compõem o pedido;
- x : coordenada x do cliente;
- y : coordenada y do cliente;

```
In [17]: import random

class Vertices:
    def __init__(self, v, p, n):
        self.v = v
        self.p = p
        self.n = n
        self.x = random.uniform(0,100)
        self.y = random.uniform(0,100)

    def __str__(self):
        return str(self.v) + " " + str(self.p) + " " + str(self.n) + "\n"

# Lista de clientes
vertices = [Vertices(random.uniform(0.001, 0.01),
                    random.randint(10,1001), random.randint(1,11))
            for i in range(0,N)]
```

```

# Os 5 primeiros clientes são centros de distribuição
for i in range(5):
    vertices[i].v = vertices[i].p = vertices[i].n = 0

```

• **Dados a respeito do veículo do tipo $k \in K$**

- V_k : volume máximo (em m^3) que o veículo pode transportar;
- P_k : valor máximo (em reais) que o veículo pode transportar;
- N_k : quantidade de veículos disponíveis;
- $(v_f)_k$: velocidade a qual o veículo se move entre o centro de distribuição e a primeira entrega, assim como entre a última entrega e o centro de distribuição;
- $(v_d)_k$: velocidade com a qual o veículo se move entre duas entregas;
- $(t_c)_k$: tempo médio para se carregar um pacote no veículo;
- $(t_d)_k$: tempo médio necessário para descarregar um pacote do veículo e entregá-lo ao cliente;
- $(p_h)_k$: custo médio por hora do veículo;
- $(p_{km})_k$: custo médio por quilômetro percorrido pelo veículo;
- $(p_f)_k$: custo fixo diário do veículo;

```

In [18]: class Veiculo:
    def __init__(self, V, P, Nv, vf, vd, tc, td, ph, pkm, pf):
        self.V = V
        self.P = P
        self.Nv = Nv
        self.vf = random.randint(vf - 5, vf + 5)
        self.vd = random.randint(vd - 5, vd + 5)
        self.tc = random.uniform(tc, 3*tc)
        self.td = td
        self.ph = ph
        self.pkm = pkm
        self.pf = pf

# Lista de informações sobre os veículos
veiculos = [Veiculo(0, 0, 0, 25, 30, 0.01, 0, 0, 0, 0) for i in range(5)]

# Tipo 0: Van
veiculos[0].V = random.randint(8,16)
veiculos[0].P = random.randint(70000,75000)
veiculos[0].Nv = random.randint(10,20)
veiculos[0].td = random.uniform(0.04, 0.08)
veiculos[0].ph = random.randint(30,60)
veiculos[0].pkm = random.randint(2,4)
veiculos[0].pf = random.randint(100,200)

# Tipo 1: Mini-Van
veiculos[1].V = random.randint(2,4)
veiculos[1].P = random.randint(70000,75000)

```

```

veiculos[1].Nv = random.randint(10,20)
veiculos[1].td = random.uniform(0.02, 0.04)
veiculos[1].ph = random.randint(30,60)
veiculos[1].pkm = random.randint(2,4)
veiculos[1].pf = random.randint(90,180)

```

Tipo 2: Comum

```

veiculos[2].V = random.uniform(0.7,1.4)
veiculos[2].P = random.randint(30000,35000)
veiculos[2].Nv = random.randint(20,30)
veiculos[2].td = random.uniform(0.02, 0.04)
veiculos[2].ph = random.randint(30,60)
veiculos[2].pkm = random.randint(1,2)
veiculos[2].pf = random.randint(60,120)

```

Tipo 3: Motocicleta

```

veiculos[3].V = random.uniform(0.02,0.04)
veiculos[3].P = random.randint(1000,5000)
veiculos[3].Nv = random.randint(20,30)
veiculos[3].td = random.uniform(0.02, 0.04)
veiculos[3].ph = random.randint(30,60)
veiculos[3].pkm = random.randint(1,2)
veiculos[3].pf = random.randint(40,80)

```

Tipo 4: Van terceirizada

```

veiculos[4].V = random.uniform(0.08,0.16)
veiculos[4].P = random.randint(75000,80000)
veiculos[4].Nv = N
veiculos[4].td = random.uniform(0.04, 0.08)
veiculos[4].ph = 0
veiculos[4].pkm = random.randint(2,4)
veiculos[4].pf = 0

```

- **Representação dos clientes**

- 0: cada centro de distribuição dentro de uma região;
- $i, j \in \{0, \dots, N\}$: centro de distribuição e clientes;
- d_{ij} : distância entre os pontos $i, j \in \{0, \dots, N\}$.

- Geração do arquivo

```
In [19]: file = open("InstanciaTeste.txt", "w") # Abrindo o arquivo
```

```

file.write(str(N) + "\n") # Número de clientes
file.write(str(R) + "\n") # Número de sub-regiões
file.write(str(K) + "\n") # Tipos de veículos

```

```

file.write(str(H) + "\n") # Carga horária diária

# Informações sobre os vértices
for v in vertices:
    file.write(str(v.x) + " " + str(v.y) + " " + str(v.v) +
               " " + str(v.p) + " " + str(v.n) + "\n")

# Informações sobre os veículos
for u in veiculos:
    file.write(str(u.V) + " " + str(u.P) + " " + str(u.Nv) +
               " " + str(u.vf) + " " + str(u.vd) + " " +
               str(u.tc) + " " + str(u.td) + " " +
               str(u.td) + " " + str(u.ph) + " " +
               str(u.pkm) + " " + str(u.pf) + "\n")

file.close()

```